1. Q: What is the purpose of your server-stats.sh script?
A: The script provides a snapshot of a Linux server's health by reporting CPU usage, memory usage, disk usage, top processes, failed login attempts, and network statistics. It helps in troubleshooting performance issues and understanding system load at a glance.

---

2. Q: How does your script calculate CPU usage?
A: It primarily uses the mpstat command if available, which reports CPU idle time. CPU usage is then calculated as 100 - idle%. If mpstat is not available, it falls back to parsing the top command's CPU statistics.

---

3. Q: How do you calculate memory usage in your script?
A: The script uses the free command to get total and used memory. It then calculates the percentage as (used / total) * 100. The script also prints memory in a human-readable format (e.g., GB/MB).

---

4. Q: How do you capture disk usage statistics?
A: The script uses df -h --total, which shows total disk usage across all mounted filesystems, including used, free, and percentage utilization.

---

5. Q: How do you identify the top processes consuming CPU and memory?
A: The script uses ps -eo pid,comm,%cpu,%mem --sort=-%cpu | head -n 6 for CPU and sorts by memory (--sort=-%mem) for memory usage. This provides the Top 5 along with their PID, command name, and usage percentages.

---

6. Q: How does your script report network usage?
A: It parses /proc/net/dev to extract RX (received) and TX (transmitted) byte counts per network interface. This gives a snapshot of how much data has passed through each interface since the system booted.

---

7. Q: How do you list active network connections in your script?
A: The script checks for the presence of ss (preferred) or netstat. It outputs established TCP connections and limits the view to the top 10 entries for readability.

---

8. Q: Why did you include failed login attempts in your script?
A: Security is an important aspect of server performance monitoring. Repeated failed login attempts may indicate brute-force attacks. By checking /var/log/auth.log or /var/log/secure, the script highlights the last 10 failed login attempts.

---

9. Q: What would you add to improve this script further?
A: Potential improvements include:

- Real-time monitoring (like bandwidth per second)
- Alerts via email/Slack when thresholds are crossed
- Historical logging of stats for trend analysis
- Integration with tools like sar, dstat, or Prometheus for advanced monitoring

---

10. Q: What are some limitations of this script compared to enterprise monitoring tools?
A: This script only provides point-in-time statistics; it doesn’t store historical data or send alerts. Enterprise tools like Prometheus, Grafana, or Nagios can do continuous monitoring, visualization, and alerting, which are better suited for production environments.